What is claimed is:

1. A method to optimize a program comprising:

cold translating a program from a first language to a second language;

determining a cold execution trip count;

inserting instructions to calculate a hot execution trip count if the cold execution trip count is less than a predetermined trip count threshold;

identifying a loop in the translated program that is a candidate for optimization using profile data;

inserting instrumentation into the loop to develop profile data; and

inserting a prefetching instruction into the loop if the profile data indicates a load instruction in the loop meets a predefined criteria.

2. A method as defined in claim 1 wherein inserting instrumentation into the loop comprises:

finding a load instruction in the loop; and

inserting a first instruction sequence to record addresses associated with the load instruction.

3. A method as defined in claim 2 wherein the first instruction sequence causes the addresses to be recorded in a buffer associated with the loop, and inserting instrumentation into the loop further comprises:

inserting a second instruction sequence into the loop to trigger processing of the addresses in the buffer to determine if the profile data indicates a load instruction in the loop meets a predefined criteria.

4. A method as defined in claim 1 wherein profile data identifies the load instruction as at least one of a single stride load, a multiple stride load, a cross stride load, and a base load of a cross stride load.

5. A method to optimize a program comprising:

cold translating the program from a first instruction set to a second instruction set;

executing the translated program;

identifying a hot loop in the translated program that meets a first predefined criteria;

gen-translating the hot loop; and

if the hot loop meets a second predefined criteria, use-translating the hot loop.

6. A method as defined in claim 5 wherein cold translating the program comprises:

identifying a block in a foreign program;

inserting instructions to update a first counter into an instruction block to determine the number of times the instruction block is executed; and

analyzing the first counter to determine if the block is a candidate for optimization.

7. A method as defined in claim 5 wherein gen-translating and use-translating the program each comprises translating the first instruction set to an intermediate instruction set and translating the intermediate instruction set to the second instruction set.

8. A method as defined in claim 7 wherein the intermediate instruction set comprises an instruction set different than the first instruction set and different than the second instruction set.

9. A method as defined in claim 5 wherein identifying the hot loop in the translated program comprises conditioning a loop by a least common specialization operation.

10. A method as defined in claim 9 wherein the least common specialization operation comprises:

identifying a block of instructions that is a least common denominator block with other loops;

rotating the loop such that the least common denominator block is a head of the loop.

11. A method as defined in claim 5 wherein identifying the hot loop in the translated program comprises:

using at least one of a cold execution trip count to determine the average number of times the hot loop is executed during cold execution or a hot

execution trip count to determine the number of times the hot loop is executed.

12. A method as defined in claim 11 wherein the cold trip count comprises instructions to determine the frequency a loop entry block is taken and the frequency the loop back edge is taken.

13. A method as defined in claim 11 wherein the hot loop is gen-translated if the hot loop contains a load instruction and a value of at least one of a hot trip count and a cold trip count is greater than a predetermined threshold.

14. A method as defined in claim 13 wherein the hot loop is only gen-translated if the load instruction does not access data in a stack or have a loop invariant load address.

15. A method as defined in claim 13 wherein the hot loop is optimized by a normal hot translation if the cold trip count is less than the predetermined threshold.

16. A method as defined in claim 5 wherein gen-translating comprises:

identifying a load instruction within the hot loop;

inserting a profiling instruction in association with the load instruction;

inserting a profiling control instruction in a loop entry block of the loop to control the number of times the load instruction is profiled;

executing the profiling instruction to profile the load instruction; and

executing the profiling control instruction to determining if the load

has been profiled more than a predetermined number of times.

17. A method as defined in claim 16 wherein the profiling instruction comprises an instruction to assign the load instruction a unique identification number and an instruction to collect profiling information.

18. A method as defined in claim 17 wherein the unique identification number is stored with a data address of the load instruction.

19. A method as defined in claim 16 wherein the profiling information comprises stride information.

20. A method as defined in claim 16 wherein the profiling control instruction comprises a counter to determine how many times the load instruction has been profiled.

21. A method as defined in claim 5 wherein use-translating comprises:

analyzing the profile information; and

inserting a prefetching instruction for the load instruction.

22. A method as defined in claim 21 further comprising eliminating redundant prefetched loads.

23. A method as defined in claim 21 wherein analyzing the profile information comprises determining if the load instruction is at least one of: a single stride load, a multiple stride load, a cross stride load; and a base load.

24. A method as defined in claim 5 further comprising linking the use-translated hot loop into the native program.

25. An apparatus to optimize a program comprising:

a cold translator to translate the program from a first instruction set to a second instruction set;

a hot loop identifier to identify a hot loop in the translated program and to determine if the hot loop should be gen-translated.;

a gen-translator to instrument the hot loop with instructions to collect profile information; and

a use-translator to optimize an instruction associated with the hot loop if the profile information determines that the hot loop should be optimized.

26. An apparatus as defined in claim 25 wherein the hot loop identifier identifies a loop as a hot loop by:

counting a number of times an instruction block associated with the loop is executed;

determining an average number of times the loop is executed; and

comparing the average number of times the loop is executed to a

predetermined threshold.

27. An apparatus as defined in claim 25 wherein the hot loop identifier identifies a hot loop in the translated program by conditioning a loop by a least common specialization operation.

28. An apparatus as defined in claim 27 wherein the least common specialization operation comprises:

identifying a block of instructions that is a least common denominator block with other loops;

rotating the loop such that the least common denominator block is a head of the loop.

29. An apparatus as defined in claim 25 wherein the gen-translator and the use-translator each translates the program from the first instruction set to an intermediate instruction set and from the intermediate instruction set to the second instruction set.

30. An apparatus as defined in claim 25 wherein the gen-translator comprises:

a load instruction identifier to identify a load instruction within the hot loop and having at least one predetermined characteristic;

a profiler to insert profiling instructions into the hot loop if the load instruction identifier identifies a load instruction within the hot loop having the at

least one predetermined characteristic.

31. An apparatus as defined in claim 30 wherein the profiler collects stride information for the load instruction.

32. An apparatus as defined in claim 25 wherein the use-translator comprises:

a profile analyzer to determine a load instruction type for the load instruction based on the profile data;

an optimizer to insert a prefetch instruction into the loop for the load instruction; and

a code linker to couple the hot loop to the program.

33. An apparatus as defined in claim 32 wherein the optimizer determines an address to be prefetched based on the load instruction type.

34. An apparatus as defined in claim 32 wherein the load instruction type comprises at least one of: a single stride load, a multiple stride load, a cross stride load, and a base load of a cross stride load.

35. A machine readable medium storing instructions structured to cause a machine to:

cold translate a program from a first language to a second language;

determine a cold execution trip count;

insert instructions to calculate a hot execution trip count if the cold execution trip count is less than a predetermined trip count threshold;

identify a loop in the translated program;

insert instrumentation into the loop to develop profile data if the hot execution trip count associated with the loop exceeds a predetermined threshold; and

insert a prefetching instruction into the loop if the profile data indicates a load instruction in the loop meets a predefined criteria.

36. A machine readable medium as defined in claim 35 wherein the load instruction comprises at least one of: a single stride load, a multiple stride load, a cross stride load, and a base load of the cross stride load.